(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 161 048 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
05.12.2001 Bulletin 2001/49

(51) Int Cl.[7]: **H04L 29/06**

(21) Application number: **01111875.9**

(22) Date of filing: **16.05.2001**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **19.05.2000 US 575330**

(71) Applicant: **Attachmate Corporation**
Bellevue, Washington 98006 (US)

(72) Inventors:
- **Hardwick, Brian Keith**
  West Harrison, Indiana 47060 (US)
- **Towne, Calvin David**
  Franklin, Ohio 45005 (US)

(74) Representative: **Grünecker, Kinkeldey,**
Stockmair & Schwanhäusser Anwaltssozietät
Maximilianstrasse 58
80538 München (DE)

(54) System and method for secure duplex browser communication over disparate networks

(57) A system and method for secure duplex browser communication over disparate networks provides duplex communication between applications such as a browser program running on a client computer system and server applications running on a server computer system. Standard web-based protocols used with the duplex communication allow use of built-in browser program features such as related to security and navigation that would otherwise be specially provided. Given the request-response nature of many of the standard web-based protocols, use of standard web-based protocols for duplex communication has not been readily attainable in the past. A duplex transport system to provide the duplex communication includes a client component running on the client computer system and a server component running on the server computer system. The browser program controls one or more browser applications configured to run on the client computer system. One or more instances of the client component and one or more instances of the server component are run to form one or more sessions each having session identifiers. Each session has one or more data pipes, which are sub-sessions. A particular data pipe has a pipe identifier and provides two independent data paths of duplex data traffic between the browser applications that are communicatively linked to the instance of the client component and the server applications communicatively linked to the instance of the server component that are both associated with the respective session of the particular data pipe. Messages of the duplex data traffic contain both session and data pipe identifiers.

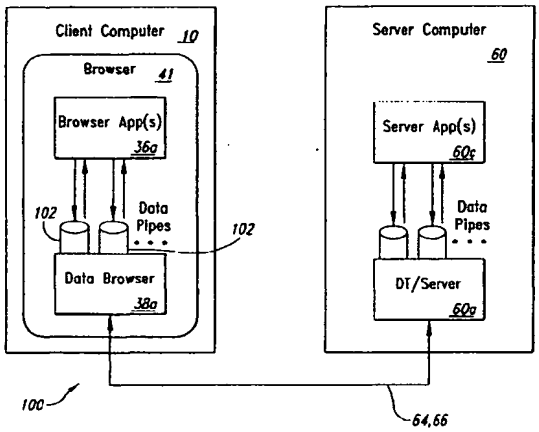

Fig. 2

EP 1 161 048 A2

the client and server computers used in the depicted embodiment of the present invention.

[0011] Figure 3 is a flowchart detailing actions involved in establishing a communication session used in the depicted embodiment.

[0012] Figures 4 - 7 are communication diagrams illustrating implementations for upstream and downstream components of data pipes used in the depicted embodiment.

DETAILED DESCRIPTION OF THE INVENTION

[0013] A browser communication system and related method for secure, duplex browser communication over disparate networks is described. In the following description, numerous specific details are provided to provide a thorough understanding of embodiments of the invention. One skilled in the relevant art, however, will recognize that the invention can be practiced without one or more of these specific details, or with other equivalent elements and components, etc. In other instances, well-known components and elements are not shown, or not described in detail, to avoid obscuring aspects of the invention or for brevity.

[0014] Figure 1 and the following discussion provide a brief, general description of a suitable computing environment in which the invention can be implemented. Although not required, embodiments of the invention will be described in the general context of computer-executable instructions, such as program application modules, objects, or macros being executed by a personal computer. Those skilled in the relevant art will appreciate that the invention can be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, mini computers, mainframe computers, and the like. The invention can be practiced in distributed computing environments where tasks or modules are performed by remote processing devices, which are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[0015] Referring to Figure 1, a conventional personal computer referred herein as a client computer 10 includes a processing unit 12, a system memory 14 and a system bus 16 that couples various system components including the system memory to the processing unit. The processing unit 12 may be any logic processing unit, such as one or more central processing units (CPUs), digital signal processors (DSPs), application-specific integrated circuits (ASIC), etc. Unless described otherwise, the construction and operation of the various blocks shown in Figure 1 are of conventional design. As a result, such blocks need not be described in further detail herein, as they will be understood by those skilled in the relevant art.

[0016] The system bus 16 can employ any known bus structures or architectures, including a memory bus with memory controller, a peripheral bus, and a local bus. The system memory 14 includes read-only memory ("ROM") 18 and random access memory ("RAM") 20. A basic input/output system ("BIOS") 22, which can form part of the ROM 18, contains basic routines that help transfer information between elements within the client computer 10, such as during start-up.

[0017] The client computer 10 also includes a hard disk drive 24 for reading from and writing to a hard disk 25, and an optical disk drive 26 and a magnetic disk drive 28 for reading from and writing to removable optical disks 30 and magnetic disks 32, respectively. The optical disk 30 can be a CD-ROM, while the magnetic disk 32 can be a magnetic floppy disk or diskette. The hard disk drive 24, optical disk drive 26 and magnetic disk drive 28 communicate with the processing unit 12 via the bus 16. The hard disk drive 24, optical disk drive 26 and magnetic disk drive 28 may include interfaces or controllers (not shown) coupled between such drives and the bus 16, as is known by those skilled in the relevant art. The drives 24, 26 and 28, and their associated computer-readable media, provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the client computer 10. Although the depicted client computer 10 employs hard disk 25, optical disk 30 and magnetic disk 32, those skilled in the relevant art will appreciate that other types of computer-readable media that can store data accessible by a computer may be employed, such as magnetic cassettes, flash memory cards, digital video disks ("DVD"), Bernoulli cartridges, RAMs, ROMs, smart cards, etc.

[0018] Program modules can be stored in the system memory 14, such as an operating system 34, one or more application programs 36, other programs or modules 38 and program data 40. The system memory 14 also includes a browser 41 for permitting the client computer 10 to access and exchange data with sources such as web sites of the Internet, corporate intranets, or other networks as described below, as well as other server applications on server computers such as those further discussed below. The browser 41 is markup language based, such as Hypertext Markup Language (HTML) and operates with markup languages that use syntactically delimited characters added to the data of a document to represent the structure of the document.

[0019] While shown in Figure 1 as being stored in the system memory 14, the operating system 34, application programs 36, other programs/modules 38, program data 40 and browser 41 can be stored on the hard disk 25 of the hard disk drive 24, the optical disk 30 of the optical disk drive 26 and/or the magnetic disk 32 of the magnetic disk drive 28. A user can enter commands and information into the client computer 10 through input devices such as a keyboard 42 and a pointing device such as a mouse 44. Other input devices can include a microphone, joystick, game pad, scanner, etc. These and

client computer 10, or a virtual machine. In the depicted embodiment, the DT/Browser 38a and the DT/Server 60a communicate using the HTTP. Security features utilized by the depicted embodiment include those specified by Internet and World Wide Web (WWW) standards organizations, such as SSL/TLS and IPSEC.

[0027] Other embodiments of the duplex transport system 100 utilize other request-response type protocols, other compatible security protocols and media for communication, and/or the same and/or other protocols approved by communications standards organizations including but not limited to such standards organizations as the International Telecommunications Union (ITU) including such committees as the Telecommunications, and the Telecommunications Standards Sector committee, and the Internet Architecture Board including such task forces as the Internet Engineering Task Force and the Internet Research Task Force.

[0028] All communication between the browser applications 36a and one of the server applications 60c is conducted through one of the data pipes 102. A DT Session is an association between an instance of the DT/Browser 38a and an instance of the DT/Server 60a. The server computer 60 can support one or more concurrent instances of the DT/Server 60a having associations through DT Sessions with one or more instances of the DT/Browser 38a existing on one or more of the client computers 10. Creation of the data pipes 102 are dependent upon creation of one or more DT Sessions.

[0029] The process of creating a DT Session starts with one of the server applications 60c registering a Session Listener callback function with the DT/Server 60a (step 112 of Figure 3). Based upon some initiating action on the client computer 10, one of the browser applications 36a creates an instance of the DT/Browser 38a to run on the client computer (step 114). Subsequently, the DT/Browser 38a establishes communication over the WAN/Internet 66 with a daemon running on the server computer 60 (step 116), which consequently causes creation of an instance of the DT/Server 60a to run on the server computer 60 (step 118). A Session Identifier that is unique to the particular DT Session is assigned (step 120) to be used in managing each DT Session created because DT Sessions may be multiplexed through a single network socket resource. The server application 60c that registered the Session Listener is then notified of the new instance of the DT/Server 60a (step 122).

[0030] Each DT Session provides one or more of the data pipes 102, which are independent duplex sub-sessions. Upon creation, each DT Session provides a first data pipe 102 referred to as the primary pipe. If more of the data pipes 102 are required, either one of the browser applications 36a or one of the server applications 60c submits requests with respect to the particular DT Session involved. To create more of the data pipes 102 in addition to the primary pipe for a particular DT Session, the server application 60c associated with the particular DT Session registers a Pipe Listener callback function with the DT/Server instance of the particular DT Session (step 124). When the browser application 36a of the particular DT Session create an instance of the data pipe 102 from the associated DT/Browser instance, a corresponding instance of the data pipe 102 from the associated DT/Server instance is also created (step 126), and the associated server application 60c is notified through the Pipe Listener callback function (step 128). Alternatively, a DT/Server instance can initiate the data pipe 102 through steps 124, 126, and 128. As a result of a DT/Server instance initiating a data pipe 102, an associated DT/Browser instance is created. If more pipes are required (yes in step 130), the procedure is repeated starting with registering another Pipe Listener (step 124). Otherwise, the procedure ends if no more pipes are required. Pipes may be closed and new ones created at any time while the DT Session is active.

[0031] Each of the data pipes 102 is assigned a Pipe Identifier that is unique to its associated DT Session. The Pipe Identifier is important because every request and reply message as part of request-reply communication between associated instances of the DT/Browser 38a and the DT/Server 60a carries multiplexed pipe traffic. Each request - reply carries message parameters including the Pipe Identifier and a Pipe Sequence Number, which identifies order sequence of messages within a particular one of the data pipes 102. The Pipe Sequence Number is used for matching requests and replies for overlapped requests (discussed further below).

[0032] The duplex transport system 100 includes three browser functions to be used with the data pipes 102 associated with the instance of the DT/Browser 38a and three server functions to be used with the data pipes 102 associated with the instance of the DT/Server 60a. The three browser functions include Browser Write, Browser Read (synchronous), and Browser Receive (asynchronous). In alternative embodiments having client applications involving duplex communication with other server applications, similar write, read, and receive functions would be utilized by the client applications. Under Browser Write, one of the browser applications 36a presents its data buffer and length. Control returns to the browser application 36a either after data has been placed in an outgoing buffer of the data pipe 102 of the associated instance of the DT/Browser 38a, after the data has been sent to the data pipe 102 of the associated instance of the DT/Server 60a, or after a reply has been received from the data pipe 102 of the associated instance of the DT/Server 60a.

[0033] Under Browser Read (synchronous), one of the browser applications 36a presents its data buffer for reading and its buffer maximum length. Data is placed in the data buffer of the browser application 36a and control returned to the browser application either when data is received from the data pipe 102 of the associated instance of the DT/Server 60a or when data exists in the

associated instance of the DT/Browser 38a receives the initial HTTP Post Reply (communication 148) causing overlapping. Pipe Sequence Numbers are used for tracking the HTTP requests and replies and are particularly helpful with the overlapping of the upstream overlapped implementation.

[0041] For server-to-client single direction data flow, the downstream components of the data pipes 102 of the DT/Browser 38a and the DT/Server 60a have a downstream basic implementation and a downstream read-ahead implementation. The downstream basic implementation starts when one of the browser applications 38a that is associated with a particular DT Session prepares to receive data from one of the server applications 60c that is associated with the same particular DT Session by invoking the Browser Read function and presenting the data buffer of the browser application to the downstream component of the data pipe 102 of the instance of the DT/Browser 38a associated with the particular DT Session (communication 160 of Figure 6).

[0042] Next the associated instance of the DT/Browser 38a sends an HTTP Get Request to the instance of the DT/Server 60a associated with the particular DT Session (communication 162). If no data is available at the instance of the DT/Server 60a associated with the particular DT Session from the associated server application 60c when the associated instance of the DT/Server 60a receives the HTTP Get Request, a timer is started with a Get Timeout value. If the timer expires before any data is available, an HTTP Get Reply with no data is sent back to the associated instance of the DT/Browser 38a causing the associated instance of the DT/Browser to re-send the HTTP Get Request. This refresh cycle is intended to keep the browser from timing out and closing the connection prematurely.

[0043] In the case illustrated in Figure 6, the associated server application 60c sends data to the data pipe 102 of the associated instance of the DT/Server 60a with a Server Write (communication 164) before timer expiration. The associated instance of the DT/Server 60a then sends a HTTP Get Reply with the data to the associated instance of the DT/Browser 38a (communication 166) and returns control to the associated server application 60c with a Server Write Return (communication 168). The data pipe 102 of the associated instance of the DT/Browser 38a then returns control to the associated browser application 36a along with the data with a Browser Read Return (communication 170).

[0044] The downstream read-ahead implementation (Figure 7) differs from the downstream basic implementation (Figure 6) in that the downstream basic implementation relies on the Browser Read function to cause an HTTP Get Request, whereas the downstream read-ahead implementation issues an HTTP Get request independently of any Browser Reads. As a consequence of this difference between the downstream basic and downstream read-ahead implementations, the order of communication for the downstream basic implementation is 160, 162, 164, 166, 168, and 170 as shown in Figure 6, whereas the order of communication for the downstream read-ahead implementation is 162, 164, 166, 168, 160, and 172 as shown in Figure 7. With the downstream read-ahead implementation (Figure 7), data is sent from the associated server application 60c through the data pipe 102 of the associated instance of the DT/Server 60a on to the data pipe 102 of the associated instance of the DT/Browser 38a (particularly communications 162, 164, and 166) before the associated browser application 36a prepares to receive data by invoking the Browser Read (communication 160).

[0045] For the downstream read-ahead implementation (Figure 7), after the Browser Read (communication 160) occurs, the data pipe 102 of the associated instance of the DT/Browser 38a sends a Browser Read Return (synchronous) along with the data to the associated browser application 36a (communication 172). The downstream read-ahead implementation has an option for the associated instance of the DT/Browser 38a of using a Browser Receive (asynchronous) to send data to the associated browser application 36a instead of a Browser Read Return for communication 172. If the Browser Receive is used, then the Browser Read in communication 160 is unnecessary. The downstream basic implementation does not have the Browser Receive (asynchronous) option. When using the Browser Read (synchronous) option, if a Browser Read (communication 160) is not outstanding when data arrives at the associated instance of the DT/Browser 38a, the data is buffered. A buffer full condition will block subsequent HTTP Get Requests from the associated instance of DT/Browser 38a until for example, a Browser Read (communication 160) is received by the associated instance of the DT/Browser 38a.

[0046] Another version of the downstream read-ahead implementation includes an overlapped feature whereas the associated instance of the DT/Browser 38a may send additional HTTP Get Requests to the instance of the DT/Server 60a associated with the particular DT Session in one or more additional communications 162. The instance of the DT/Server 60a associated with the particular DT session queues each HTTP Get request until data is available from additional Server Write data calls (additional communications 164). This causes an overlapping of the communication wherein pipe sequence numbers are used to track the overlapping.

[0047] From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims.

8. A duplex transport system for use with a client computer system having a client application controlling a utility application, the client computer system communicatively linked to a network system and a server computer system having a server application, the server computer system communicatively linked to the network system, the duplex transport system comprising:

   a client component configured to run as an instance on the client computer system, the instance of the client component being communicatively linked to one of the utility applications;
   a server component configured to run as an instance on the server computer system, the instance of the server component being communicatively linked to one of the server applications; and
   the client component and the server component configured such that the instance of the client component is associated with the instance of the server component in an association to form a session, the session having a session identifier and a sub-session designated as a data pipe, the data pipe having a pipe identifier and configured to provide two independent data paths of duplex data traffic between the utility application communicatively linked to the instance of the client component and the server application communicatively linked to the instance of the server component.

9. The duplex transport system of claim 8 wherein the client computer and the server component are further configured such that the duplex data traffic of the data pipe of the session formed from the association between the instance of the client component and the instance of the server component utilizes Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), Internet Protocol Secure (IPSEC), Secure Sockets Layer/ Transport Layer Security (SSL/TLS), other request-response protocols, and/or the same and/or other protocols approved by communication standards organizations including but not limited to such standards organizations as the International Telecommunications Union (ITU) including such committees as the Telecommunications, and the Telecommunications Standards Sector committee, and the Internet Architecture Board including such task forces as the Internet Engineering Task Force and the Internet Research Task Force.

10. The duplex transport system of claim 8 wherein the client computer and the server component are further configured such that the data pipe of the session formed from the association between the instance of the client component and the instance of the server component provides the data paths of duplex data traffic comprising messages that each contain the pipe identifier.

11. The duplex transport system of claim 8 wherein the client computer and the server component are further configured such that the data pipe of the session formed from the association between the instance of the client component and the instance of the server component data pipe is configured to provide data paths of duplex data traffic comprising messages that each contain the pipe identifier identifying the data pipe and a pipe sequence number, the pipe sequence number identifying an order of the messages in the duplex data traffic associated with the data pipe.

12. The duplex transport system of claim 8 wherein the client computer and the server component are further configured such that the session formed from the association between the instance of the client component and the instance of the server component further comprises a second data pipe being a second sub-session of the session, the second data pipe having a pipe identifier, configured to provide two additional independent data paths of a second duplex data traffic between the utility application and the server application, and being a secondary data pipe.

13. The duplex transport system of claim 8 wherein the client component is configured to run with a browser program.

14. The duplex transport system of claim 8 wherein the client component and the server component are further configured to run as second instances where the second instances of the client component and server component are associated in an association to form a second session having a session identifier.

15. A client computer system for use with a duplex transport system and a server computer system having a server application, the client computer system and the server computer system having a server component communicatively linked to a network system, the client computer system comprising:

   a client computer;
   a browser program configured to run on the client computer, the browser program having built-in features associated with communication protocols used by the duplex transport system;
   one or more browser applications configured to run on the client computer under control of the browser program;

Task Force and the Internet Research Task Force

22. The server computer system of claim 20 wherein the server component is further configured to be associated with the client component in an association to form a session that has more than one data pipes having duplex data traffic where each message of the duplex data traffic is assigned the pipe identifier corresponding to the data pipe used by each message.

23. The server computer system of claim 20 wherein the server component is further configured to be associated with the client component in an association to form a session that has one or more data pipes that utilize the communication protocols associated with the built-in features of the browser program for the duplex data traffic.

24. The server computer system of claim 20 wherein the built-in features of the browser program involve one or more of the following: uniform resource locators (URLs), firewall/proxy navigation under Hypertext Transfer Protocol (HTTP), proxy configuration of the browser program, HTTP authentication, Transmission Control Protocol/Internet Protocol (TCP/IP), Secure Sockets Layer/Transport Layer Security (SSL/TLS), HTTP Secure (HTTPS), Internet Protocol Secure (IPSEC), and access to client certificates for use with security protocols.

25. A method for establishing duplex communication between a browser application running under control of a browser program on a client computer system and a server application running on a server computer system over a network, the method comprising:

registering a session listener callback function for the server application with a server component running on the server computer system;
initiating through the browser application creation of an instance of a client component to run on the client computer system;
establishing through the instance of the client component communication over the network with the server computer system;
based upon establishing communication between the client component and the server computer system, creating an instance of a server component to run on the server computer system;
notifying the server application through the session listener callback function of the establishment of the instance of the server component;
establishing an association between the instance of the client component and the instance of the server component as a session and assigning a session identifier to the session;
designating a sub-session of the session as a data pipe of duplex data traffic between the browser application and the server application; and
assigning a pipe identifier to the data pipe to be used by messages being sent through the data pipe.

26. The method of claim 25, further comprising:

registering a pipe listener callback function with the instance of the server component;
creating an instance of a second data pipe through the browser application from the instance of the client component and the instance of the server component; and
notifying the server application through the pipe listener callback function of creation of the second data pipe.

27. A method of transmitting data from a client computer system to a server computer system, the method comprising:

invoking a Read function through a server application on the server computer system, the server application associated with a session between an instance of a client component running on the client computer system and an instance of a server component running on the server computer system;
presenting a data buffer of the server application to an upstream component of a data pipe associated with the instance of the server component;
writing data from a browser application on the client computer system to an upstream component of a data pipe associated with the instance of the client component;
sending an Hypertext Transfer Protocol (HTTP) Post along with data to the instance of the server component; and
sending from the instance of the server component either a Server Read Return or a Server Receive callback along with the data to the server application.

28. The method of claim 27, further comprising:

sending an HTTP Post Reply to the instance of the client component; and
sending a Browser Write Return to the browser application.

29. A method of transmitting data from a server computer system to a client computer system, the method comprising:

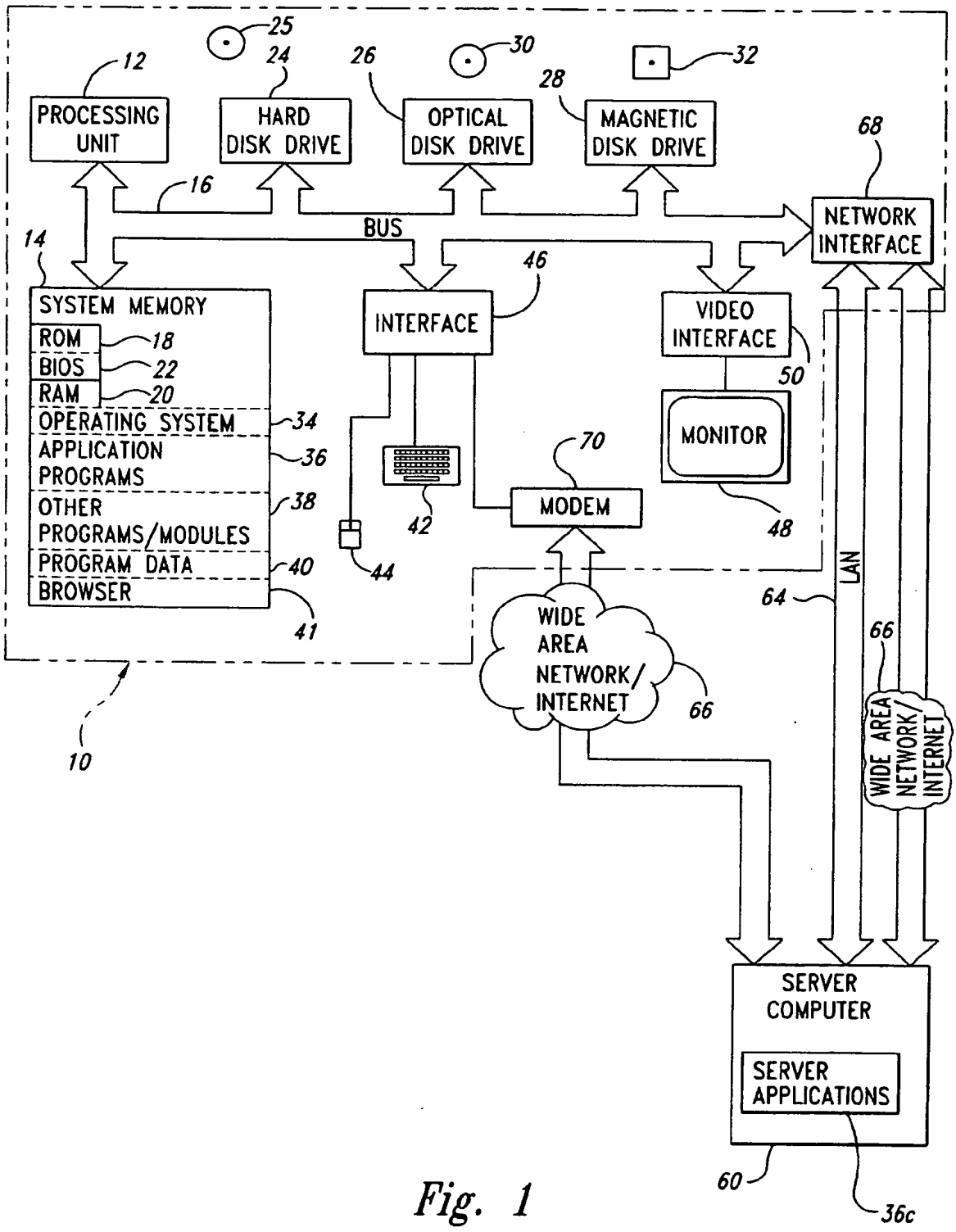

Fig. 1

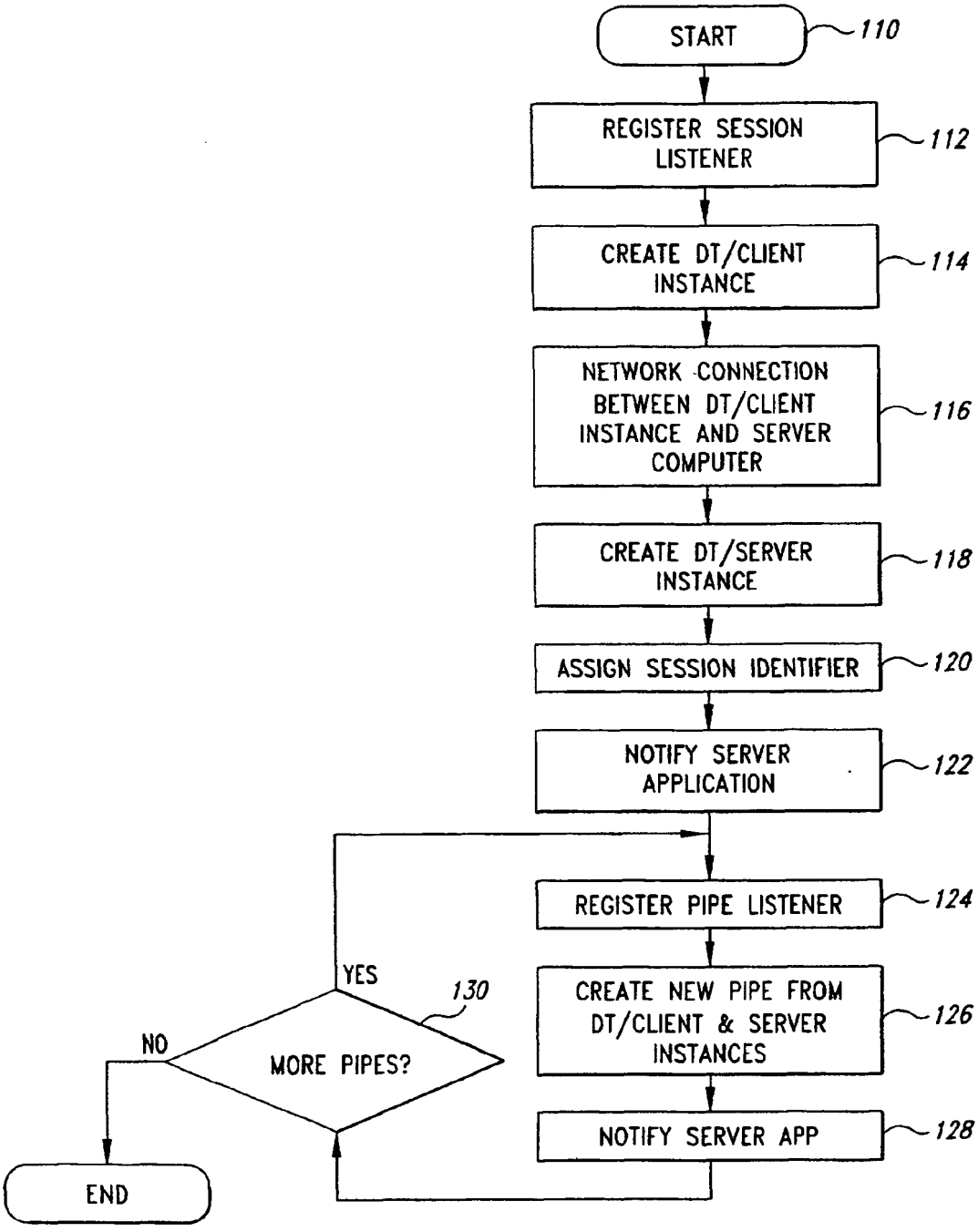

Fig. 3

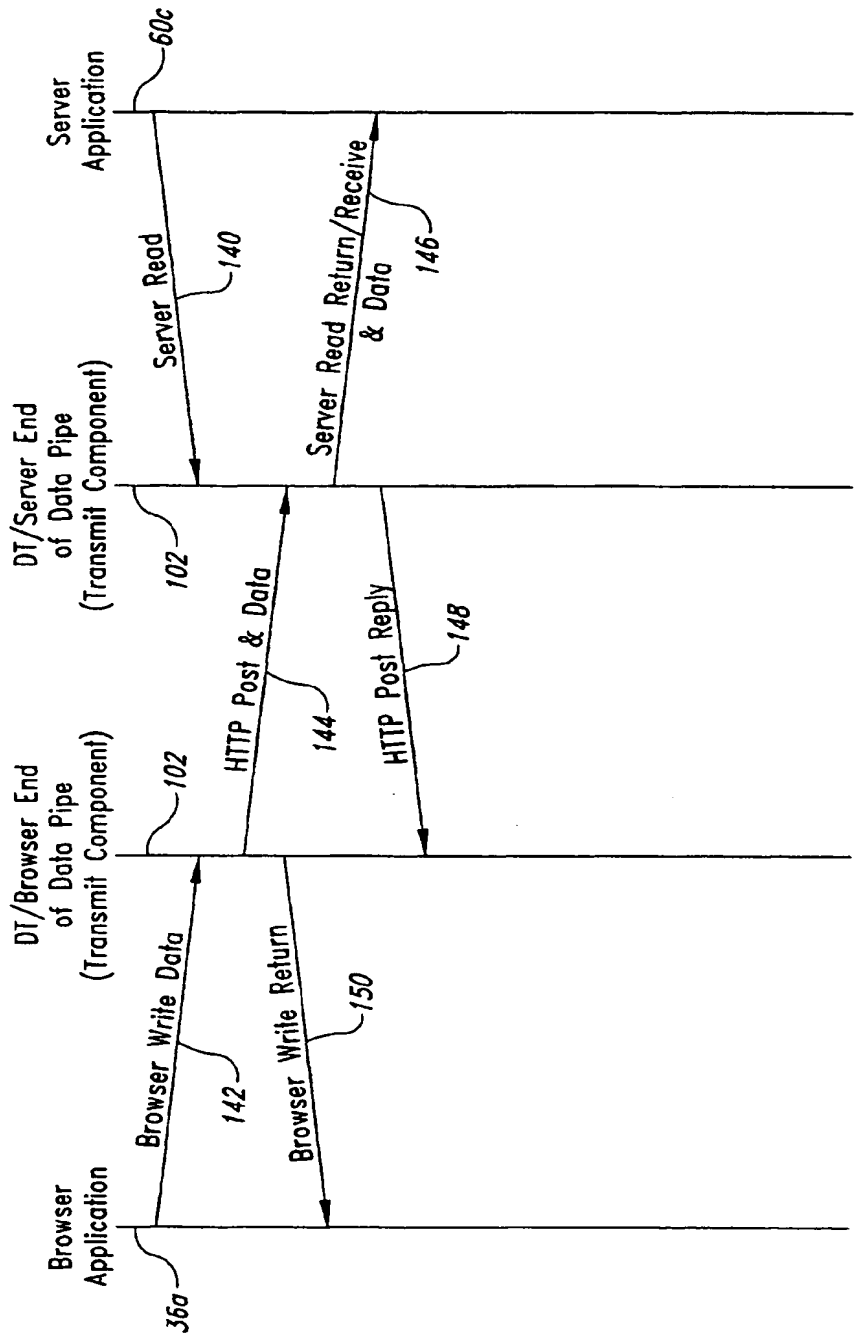

Fig. 5

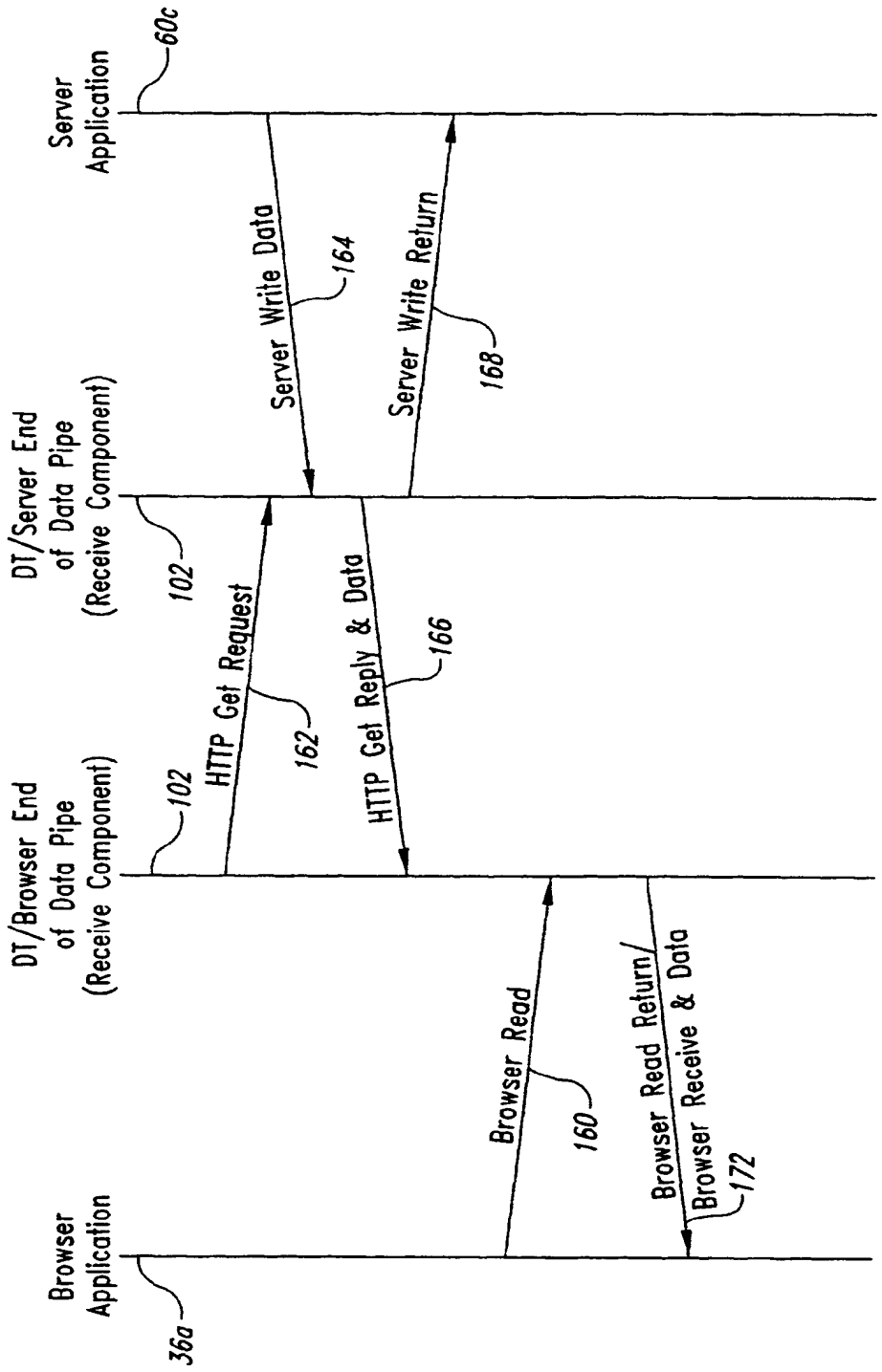

Fig. 7

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 161 048 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
16.02.2005 Bulletin 2005/07

(43) Date of publication A2:
05.12.2001 Bulletin 2001/49

(21) Application number: **01111875.9**

(22) Date of filing: **16.05.2001**

(51) Int Cl.[7]: **H04L 29/06**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **19.05.2000 US 575330**

(71) Applicant: **Attachmate Corporation**
Bellevue, Washington 98006 (US)

(72) Inventors:
- **Hardwick, Brian Keith**
  West Harrison, Indiana 47060 (US)
- **Towne, Calvin David**
  Franklin, Ohio 45005 (US)

(74) Representative: **Grünecker, Kinkeldey,**
Stockmair & Schwanhäusser Anwaltssozietät
Maximilianstrasse 58
80538 München (DE)

(54) System and method for secure duplex browser communication over disparate networks

(57) A system and method for secure duplex browser communication over disparate networks provides duplex communication between applications such as a browser program running on a client computer system and server applications running on a server computer system. Standard web-based protocols used with the duplex communication allow use of built-in browser program features such as related to security and navigation that would otherwise be specially provided. Given the request-response nature of many of the standard web-based protocols, use of standard web-based protocols for duplex communication has not been readily attainable in the past. A duplex transport system to provide the duplex communication includes a client component running on the client computer system and a server component running on the server computer system. The browser program controls one or more browser applications configured to run on the client computer system. One or more instances of the client component and one or more instances of the server component are run to form one or more sessions each having session identifiers. Each session has one or more data pipes, which are sub-sessions. A particular data pipe has a pipe identifier and provides two independent data paths of duplex data traffic between the browser applications that are communicatively linked to the instance of the client component and the server applications communicatively linked to the instance of the server component that are both associated with the respective session of the particular data pipe. Messages of the duplex data traffic contain both session and data pipe identifiers.

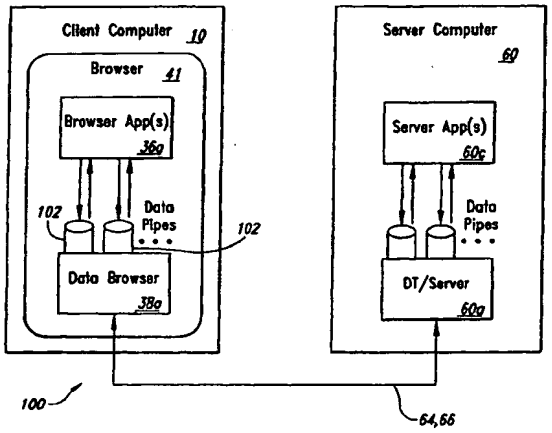

Fig. 2

EP 1 161 048 A3

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 01 11 1875

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

27-12-2004

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5999979 A	07-12-1999	EP 0956686 A1	17-11-1999
		JP 3498746 B2	16-02-2004
		JP 2000509592 T	25-07-2000
		WO 9834385 A1	06-08-1998
		EP 0956702 A1	17-11-1999
		JP 2001527709 T	25-12-2001
		WO 9834405 A1	06-08-1998
		US 6754715 B1	22-06-2004
		US 6014706 A	11-01-2000
		US 6230172 B1	08-05-2001
WO 9964958 A	16-12-1999	US 6289461 B1	11-09-2001
		AU 4558699 A	30-12-1999
		CA 2334971 A1	16-12-1999
		EP 1125207 A1	22-08-2001
		JP 2002517857 T	18-06-2002
		WO 9964958 A1	16-12-1999
		US 2001056547 A1	27-12-2001
EP 0690599 A	03-01-1996	CA 2150062 A1	31-12-1995
		CN 1117616 A	28-02-1996
		EP 0690599 A2	03-01-1996
		JP 8051468 A	20-02-1996
		SG 33392 A1	18-10-1996

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82